Tanta University- Third Year –Computer and Control Department
Dr. Hatem-2016-Normalisation

Question One

5) Define each of the following terms: Transitive dependences, Normalization and Composite key.

2- Give one example for the following relations

- Binary relationship - Ternary relationship - Unary relationship

2) The following table contains sample data for parts and for vendors who supply those parts. Perform the following activates at the table:

a- Convert the relations in this table into the first normal form.

b- Perform the second normal form to the previous resulted relations.

c- Develop a set of third normal form at the resulted relations.

Part Number	Description	Vendor Name	Address	Unit Cost
1234	Logic chip	Fast Chips Smart Chips	Intel IBM	10.00 8.00
5678	Memory chip	Fast Chips Quality Chips Smart Chips	Intel Phoenix IBM	3.00 2.00 5.00

Question Two

A engineering consultancy firm supplies temporary specialized staff to bigger companies in the country to work on their project for certain amount of time. The table below lists the time spent by each of the company's employees at other companies to carry out projects. The National Insurance Number (NIN) is unique for every member of staff.

NIN	Contract No	Hours	Employee Name	Company ID	Company Location
616681B	SC1025	72	P. White	SC115	Belfast
674315A	SC1025	48	R. Press	SC115	Belfast
323113B	SC1026	24	P. Smith	SC23	Bangor
616681B	SC1026	24	P. White	SC23	Bangor

a) Explain in which normal form this table is
b) Find the Primary Key for this relation and explain your choice.
c) Find the **Fully Functional Dependencies** on the PK and the **Partial Dependencies** on the PK.
d) Normalise the table to 2NF
e) Find the transitive dependencies on the 2NF tables
f) Normalise the tables to 3NF – Express the tables in DBML language and show the PK and FK in all the relations.

Solutions

Definitions

UNF definition: A relation (table) that contains one or more repeating groups.

1NF definition: A relation (table) in which the intersection of each row and column contains one and only one value.

2NF definition: A relation that is in 1NF and every non-primary key attribute is fully functional dependent on the primary key.

3NF definition: A relation that is in 1NF and 2NF, and in which no non-primary key attribute is transitively dependent on the primary key

Functional Dependency: Describes the relationship between attributes in a relation. For Example if A and B are attributes of a relation R, B is functionally dependent and A (denoted A→ B), if each value of A is associated with exactly one value of B. (A and B may each consist of one or more attributes)

Fully Functional Dependency: Indicates that if A and B are attributes of a relation, B is fully functionally dependant on A if B is functionally dependant on A but not on any proper subset of A.

Transitive Dependency: A condition where A,B and C are attributes of a relation such that if A→ B and B→ C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

a) and b)

The table in the example is at least in 1NF (to check for 2NF the primary key needs to be defined.) Candidate attributes are NIN, Contract No and Company ID), we can choose NIN and ContractNo as the primary key because this two will be the minimum attributes required to uniquely identify each row in the table and also will reflect better the objective of the company which is to keep track record of staff supplied in a contract to another company.

Given the PK: NIN, ContractNo each non-primary key attribute needs to be checked for fully functional dependency on the PK:

Employee name is fully functionally dependant on NIN (NIN → Employee Name). but as NIN is part of the PK, Employee Name is not fully functional dependent on the PK therefore the form is NOT on 2NF.

StaffContract (NIN, ContractNo, hours, EmployeeName, CompanyID, CompanyLocation)

Primary keys are underlined.

c)
We need to find dependencies on the PK and on parts of the PK (partial dependencies):

NIN, ContractNo → (primary key)
NIN → (partial dependency on the PK)
ContractNo → (partial dependency on the PK)
Examining each of the remaining attributes, we determine what is the dependency associated with it:

Hours: This is the main attribute that needs to be recorded – it is functionally dependant on the employee and the associated contract (the whole PK):

Employee Name: As stated previously, this is fully functionally dependent on NIN, but not on ContractNo, therefore it is a partial dependency.

Company Id and Company Location: These attributes are dependant on the ContractNo, but are not related to NIN, therefore they are partially dependant on the PK.

NIN, ContractNo → hours (primary key)
NIN → Employee Name (partial dependency on the PK)
ContractNo → CompanyId, CompanyLocation (partial dependency on the PK)

d) To normalise the table to 2NF, we need to remove the partial dependencies from the table and locate them in a new table.

StaffContract (NIN*, ContractNo*, hours)
Staff Details (NIN, EmployeeName)
ContractDetails (ContractNo, CompanyID, CompanyLocation)

e) To find transitive dependencies we need to look at the non-primary key attributes. In this case the location of the company (CompanyLocation) is functionally dependent on CompanyID; therefore it is transitively dependent on the PK for the ContractDetails relation in the 2NF tables:

ContractNo → CompanyID.
CompanyID → CompanyLocation.

f)

To normalise to 3NF this transitive dependency needs to be put in a new table with a copy of the determinant:

The ContractDetails table is divided in two more tables

Contracts (ContractNo, CompanyID*)
Company (CompanyID, CompanyLocation)

And the previous tables are left the way they are because they are 3NF already.

The final tables normalised to 3NF are:

Staff Details (NIN, EmployeeName)
StaffContract (NIN*, ContractNo*, hours)
Contracts (ContractNo, CompanyID*)
Company (CompanyID, CompanyLocation)

Primary Keys are underlined, foreign keys are marked with an (*).

Using DBML for the first set we can express the relations:

Staff Details {NIN, EmployeeName}
Primary Key: NIN
Foreign Key: None

Staff Contract {NIN, ContractNo, Hours}
Primary Key: NIN, ContractNo
Foreign Key: NIN **references** StaffDetails(NIN)
Foreign Key: ContractNo **references** Contracts(ContractNo)

Contracts {ContractNo, CompanyID}
Primary Key: ContractNo
Foreign Key: CompanyID **references** Company(CompanyID)

Company {CompanyID, CompanyLocation}
Primary Key: CompanyID
Foreign Key: None

Note:

The foreign keys selection on the tables is arbitrary, once the full tables are normalised to 3NF, the FK are used to link the tables. The previous 3NF relations could be also written as:

Staff Details (NIN*, EmployeeName)
StaffContract (NIN, ContractNo, hours)
Contracts (ContractNo*, CompanyID*)
Company (CompanyID, CompanyLocation)